REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
AUG 2011	Journal Article (Post Print)	JUN 2010 – JUL 2010

4. TITLE AND SUBTITLE

OBJECT DISCOVERY, IDENTIFICATION AND ASSOCIATION

5a. CONTRACT NUMBER
FA8750-09-2-0157 / B315SUNY

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Vijay Kumar, James Metzler, Mark Linderman, Jon Jones, Mark Alford, Adnan Bubalo, and Maria Scalzo

5d. PROJECT NUMBER
558S

5e. TASK NUMBER
IH

5f. WORK UNIT NUMBER
NC

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Missouri-Kansas City
5100 Rockhill Road
Kansas City, MO 64110

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/RISD
525 Brooks Road
Rome, NY 13441

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2011-18

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA #: 88ABW-2010-4766, DATE CLEARED: 1 SEP 2010.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Tracking process captures the *state* of an object. The state of an object is defined in terms of its dynamic and static properties such as location, speed, color, temperature, size, etc. The set of dynamic and static properties for tracking very much depends on the agency who wants to track. For example, police needs different set of properties to tracks people than to track a vehicle than the air force. The tracking scenario also affects the selection of parameters. Tracking is done by a system referred to in this paper as "Tracker." It is a system that consists of a set of input devices such as sensors and a set of algorithms that process the data captured by these input devices. The process of tracking has three distinct steps (a) object discovery, (b) identification of discovered object, and (c) object introduction to the input devices. In this paper we focus mainly on the object discovery part with a brief discussion on introduction and identification parts. We develops a formal tracking framework (model) called "Discover, Identify, and Introduce Model (DIIM)" for building efficient tracking systems. Our approach is heuristic and uses reasoning leading to learning to develop a knowledge base for object discovery. We also develop a tracker for the Air Force system called N-CET.

15. SUBJECT TERMS
N-CET, *MinSet*, Sensor, Dynamic property, Static property, Contextual signature.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			JAMES METZLER
U	U	U	UU	11	19b. TELEPONE NUMBER (*Include area code*) N/A

Object Discovery, Identification and Association

Vijay Kumar[*], James Metzler[+], Mark Linderman[+], Jon Jones[+], Mark Alford[+], Adnan Bubalo[+], and Maria Scalzo[+]
*University of Missouri-Kansas City, Kansas City, Missouri
+Air Force Research Lab, Rome, NY, USA.

Abstract

Tracking process captures the *state* of an object. The state of an object is defined in terms of its dynamic and static properties such as location, speed, color, temperature, size, etc. The set of dynamic and static properties for tracking very much depends on the agency who wants to track. For example, police needs different set of properties to tracks people than to track a vehicle than the air force. The tracking scenario also affects the selection of parameters. Tracking is done by a system referred to in this paper as "Tracker." It is a system that consists of a set of input devices such as sensors and a set of algorithms that process the data captured by these input devices. The process of tracking has three distinct steps (a) object discovery, (b) identification of discovered object, and (c) object introduction to the input devices. In this paper we focus mainly on the object discovery part with a brief discussion on introduction and identification parts. We develops a formal tracking framework (model) called "Discover, Identify, and Introduce Model (DIIM)" for building efficient tracking systems. Our approach is heuristic and uses reasoning leading to learning to develop a knowledge base for object discovery. We also develop a tracker for the Air Force system called N-CET.

Keywords: N-CET, *MinSet*, Sensor, Dynamic property, Static property, Contextual signature.

1. INTRODUCTION

Tracking process discovers the *state* of an object in real time by collecting a set of its *dynamic* properties (*DP*) and *static* properties (*SP*) and their values (instances of *DP* and *SP*.) An object is tracked by a *tracker*. A tracker is a system that is composed of a set of input devices, i.e., sensors, and a set of algorithms. A sensor has no intelligence it is purely an input device that can only capture values of a specific parameter of its surroundings. It neither has data processing capability nor can it capture any other parameter value. As a result, it cannot recognize the object it is dealing with. For example, a video camera can capture only the color of the object, it cannot record the distance and it cannot distinguish the difference between a car and a building. Similarly, radar can capture distance but not the color or any other property of the object it tracks. There are some sensors (multi-mode) that can capture values of temperature, height, location, etc. Aircrafts are usually equipped with multi-mode sensors that continuously collect more than one type of parameter values. We formally define a sensor as follows:

Definition 1. *A sensor is a data capturing device that is built to capture a specific type of data about its surroundings. A sensor has no data processing capability; as a result, it cannot differentiate between two objects.*

A tracker may need more than one sensor to capture a set of different types of data for object *discovery* and *identification*. For example, a police in a helicopter may need to use a sensor to capture speed of a car, a sensor to capture its color, and a sensor to capture its location. These three are complementary sensors because the data streams they generate can be meaningfully fused to identify the car. The inclusion of a sensor that measures humidity or temperature may not be useful in car identification so it would not be a complementary sensor according to our definition. Identification of complimentary set of sensors for gathering information at a particular situation is important in composing a node for tracking. We deal with this issue in our future work.

Definition 2. *Two sensors are complementary if the values captured by one sensor can be meaningfully fused with the data captured by the other sensor in discovering or identifying the object.*

We refer to a sensor or a set of sensors (single or multi-node) as a *node* and use this term in place of sensor. Irrespective of the type of sensors (multi-mode or single mode) they continuously capture and send the data stream to a server for object discovery.

Signal Processing, Sensor Fusion, and Target Recognition XX, edited by Ivan Kadar,
Proc. of SPIE Vol. 8050, 805016 · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.882178

Definition 3. *A node is a set of complementary sensors. Node = {s_1, s_2, … s_n} where s_i and s_j are a set of complementary sensors.*

Definition 4. *A tracker is a system that comprises a set of nodes and data processing software. Tracker = <M, {n_1, n_2, …, n_n}> where M is the software module and n_i's are nodes.*

Tracking can be defined in terms of a set of operations with the order: Find → Fix → Track → Target engage → Assess. The result of asses step defines the success or failure of trsacking. We express the first four tasks in terms of three steps (a) discovery, (b) identification and (c) association. A desired object for tracking must first be *discovered* (found) among a set of objects. For example, if the objective is to track a particular mobile vehicle in a big parking lot with full of people, stalls, etc., then first the mobile vehicle must be discovered (found). Similarly if the task is to track the movement of a terrorist cell then first it must be discovered (found) at the location. The result of the discovery may be one or a set of identical objects. In the identification (fix) step the desired object from the set of discovered objects is identified. Finally, the identified object is associated with an appropriate node (track and engage) for tracking. In air surveillance scenario a set of UAVs may engage in object discovery such as discovering terrorist cells for identification and tracking. Further, routinely radar on a helicopter or low flying plane try to discover speeding vehicles to maintain accident-free traffic. In real-world events (military, mission-critical tasks, battle front, etc.) object discovery, identification and association are continuously deployed. Values of a set of dynamic and static properties of the object, for example, location, speed, size, color, etc., determines its state. We refer to *property-value* pair as an *instance* of the property. We, therefore, define a tracking operation as a process of collecting the relevant set of instances of its *DP*s and *SP*s that can define its state any time during its existence. We define a Universal Property Set (UPS) from where a set desired *DP*s and *SP*s for an object is drawn.

The last step "assess" is the final result of data fusion and may indicate success or failure or something else. We do not include this in our investigation; however, our objective is that discovery, identification, and association should lead to successful tracking.

Definition 5. Dynamic Properties (*DP*): *It is a countably infinite set of properties whose values could change with time. DP = {dp1, dp2, ..., dpn, T} where dpi = is a dynamic property with a value domain and T = time when the dpi's value is measured.*

Examples: Reflected wave properties, speed, color, owner name, *location*, age, etc. are examples of dynamic properties of an object. Each property is associated with a value domain. The value of a dp_i comes from its domain. The size of the domain depends on the object and the organization that uses the object. For example, the average age of a person in USA is about 80 years. When the person is employed in a company then age domain could be (18 through 68) that means the company does not employ a person less than 18 years old and the person must retire from the company at the age of 68 years.

Definition 6. Static Properties (*SP*): *It is a countably infinite set of properties whose values never change with time. SP = {sp1, sp2, ..., spn, e} where spi = is a static property with a property domain.*

Examples: Manufacturing date, social security number, repair date, *location*, etc. are examples of static properties of an object. Each property is associated with a value domain.

Some members of *DP* and *SP* could be the same. For example *location* of a house is a *SP* and the location of a car is *DP*. Thus, $\{DP\} \cap \{SP\} \neq \varnothing$. In some cases the value of *location* serves as its initial value and validates the value of *location* . For example, if the initial value of *location* is Rome, NY and its next value appears as San Francisco when time is 10 minutes then one or both values are incorrect. Usually the initial value of *location* is correct. Note that *location* of a mobile object is different from *SP* in the sense that it is capable of taking different values. We make a finer distinction for some *SP*s and *DP*s. The values of one type such as the *location* of a house, density of gold, manufacturing date, etc. are invariant but some may change with time. We consider the time duration in defining the second type of *SP*. If a property does not change within a defined time period then it is regarded as *SP* otherwise it is a *DP*. For example, normally the shape of an object is not likely to change during the discovery period.

Definition 7. **Domain (D) of a property (SP or DP)***: It is a set of permissible values of a property with reference to the context. D = {pi, [vi, vj]} where vi = the minimum value vj = the maximum value of property pi.*

Definition 8. Instance (I): *We define an instance of a property as I = <p, v, e, T> where p ∈ {DP ∪ SP}, v ∈ D is the value of p, e is a margin of error, and T is time of measurement of v.*

The margin of error (*e*) is a property of the instrument that measures the value of the property. An instrument (sensor, scale, etc.) has inherent measuring limitation.

Definition 9. Universal Property Space (UPS): *It is a set of DP and SP. UPS = {DP, SP}. A subset of DP and a subset of SP of an object are drawn from here.*

Definition 10. State: *The state of an object is a complete history in terms of the subsets of I, context, and time. S = {Sid, <SE1, p1>, <SE2, p2>, …, <SEn, pn>, T}, where Sid = state identity, SEi is sensor identity, pi ∈ {DP ∪ SP}, and Ti is the time the property value was measured.*

Definition 11. Tracking: *Tracking is a process of collecting a subset of the instances such that together these subsets uniquely identify the object at any instant of time T. Thus, Tracking = <Tid, Si> where Tid = tracking process identity, Si = state identity.*

When we have insufficient information about an object then we ask the question "*what additional set of DP or SP or both is required to accurately discover the object?*" A related question is "*what combination of sensors would be required to capture the desired set of DP and SP?*" In reality we usually need to discover the object because only a partial (incomplete) set of *DP* and *SP* is available. In this paper we investigate the first question and develop efficient solution. We report the result of our investigation of the second problem, i.e., "*what combination of sensors (node) would be required to capture the desired set of DP and SP?*" in our future work.

The paper is organized as follows. Section 2 introduces the reference platform for algorithm development and presents the concepts of MinSet. Section 3 introduces a high level description of our object discovery model called DIIM and our heuristic based algorithm. Section 4 briefly covers the identification and introduction processes. Section 5 presents a model of our knowledge base and finally Section 6 concludes the paper.

2. OBJECT DISCOVERY, IDENTIFICATION AND INTRODUCTION

A reference system is required for developing our tracking framework and scheme. The algorithms that we develop will not be for the reference system; we just use it to validate our scheme. The reference system is called N-CET (Network-Centric Exploitation and Tracking) [2]. It is an experimental system to process real-time sensor data stream. Although it is a distributed system that is capable of processing all kinds of data such as network monitoring data, process synchronization data, and so on, it is actually designed to process any kind of stream data. The software module that is responsible for providing necessary data processing support is referred to as Joint Battleship Infosphere (JBI) system [3]. It is basically a Service Oriented Architecture (SOA) based system and the objective is to make JBI a global system capable of morphing itself to a most appropriate sub-system for any environment and situation for providing efficient data processing services. An offshoot of JBI is referred to as Phoenix that is being developed to satisfy the needs of a smaller group of users.

We concentrate on the aspect of N-CET that provides support for real-time stream data processing. Stream data is generated by sensors that are deployed to continuously monitor an object (human, car, house, etc.) This kind of data significantly differs from common data stored in a conventional database system in terms of their flow, sequence, contextual properties and validity duration. Data segments in a data stream could be repetitive, for example, a video recording a stationary object from a constant angle. For these reasons, stream data processing requires a different processing approach. A sensor continuously captures the data and sends it to some server for processing. Our sensor here is N-CET that monitors an object and captures the relevant set of the instances of the object.

We encounter two possible initial tracking conditions (a) complete object description (necessary instances) is available and (b) a brief high level description (only a small subset of instances) is provided. We introduce the concept of "Local Resources (*LS*)" to explain these scenarios. *LS* is the source of first input for tracking. When an event occurs then the information about this event (location, type of event, etc.) is disseminated by people who witness it. For example, when a traffic accident occurs, the local resources (i.e., other drivers or pedestrians) call 911 and provide initial set of the instances.

In reality a complete description to begin tracking is rarely available. For example, LS may inform the police the color of a suspicious car, location of an accident, fighting sound from a neighbor's house, etc. In order to begin tracking more information about the object must be discovered.

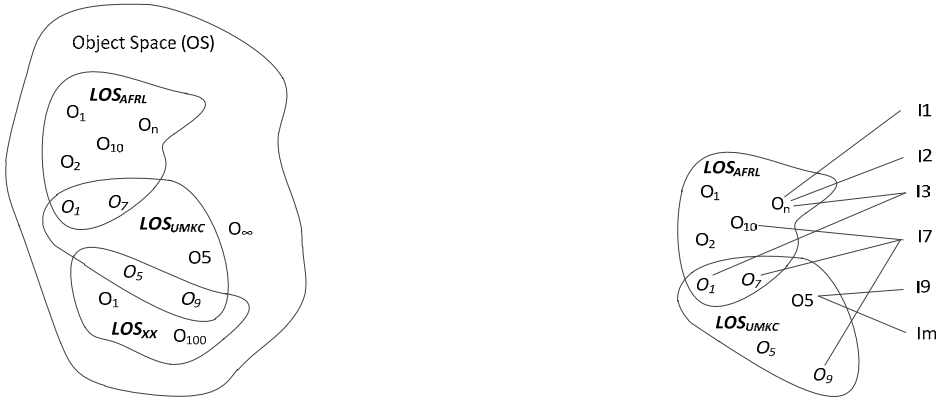
2.1 A Complete Object Description

We define a complete object description through mapping of a subset of *DP*s and *SP*s. Figures 1a and 1b illustrate the logical configuration of OS, LOS and property mapping.

An organization, for example AFRL or UMKC, can be defined as a set of objects drawn from OS. Any two LOSs share at least one DP or SP or both.

Definition 11. Object Space (OS): *It is an infinite set of objects where each object has a subset of DP and a subset of SP. OS = {<o1, DPo1, SPo1>, <o2, DPo2, SPo2> ..., <o∞, DPo∞, SP o∞>} where oi is an object and DPoi, SPoi are dynamic and static properties associated with o1.*

Definition 12. Local Object Space (LOS)*: It is a finite set of objects that defines an institution. LOS = <LOS$_i$, {<o1, DPo1, SPo1>, <o2, DPo2, SPo2>, …, <om, DPom, SPom>, I$_i$} where I$_i$ is the identity of the institution. LOS ⊂ OS.*



a b

Figure 1. Object Space OS, Local Object Space (LOS) and Property Mapping

We define a mapping function that maps a subset of *I* (instances) to an element of LOS (Figure 1b). A complete mapping creates a unique object signature that we refer to as *context signature.* A context signature uniquely identifies an object that can be then associated to a tracker for tracking to begin. Note that *LOSx ∩ LOSy ≠ {∅}; x≠y*. A *context signature* can be created only if complete information about an object is available. This is rarely the case. A tracker usually gets minimum information from local resources and builds upon this a *Minimum Set (MinSet)* for object discovery. The cardinality of *DP*, *SP*, and OS is infinite so a formal approach for composing a *MinSet* is a NP-hard problem. Our solution approach is therefore heuristic. We make a realistic assumption without any loss of generality: *our object space is finite and includes meaningful objects an organization is interested in.*

In a complete object description the discovery and identification phases are already completed by the information provider. In this scenario, the set of instance is sufficient to associate the object to a node for tracking to begin. The following example illustrates an instance of a complete object description.

Type (SP) = Car
Model (SP) = Lexus
Color (SP) = Red
Location (DP) = Intersection of Floyd St and E. Thomas St.
License plate number = OQX 999.

We introduce the concept of a *MinSet* that contains sufficient information for object discovery through this example. If the *MinSet* is incomplete then our system will be able to complete the *MinSet* to discover the object.

Definition 13. Minimum Set (*MinSet***)**: *A MinSet is a set of instances that is sufficient to discover the object. MinSet = <$MinSet_{id}$, min(I), T> where min(I) indicate minimum number of instances for object discovery, Tid is the identity of MinSet and T indicates the time the MinSet was created*.

2.2 Composition of a Minimum Set (*MinSet)*

This is a relatively hard problem. We encounter two situations (a) no information is available and (b) *some* information is available. The first situation presents two cases (i) no information is available and there is no possibility of obtaining any information and (ii) no information is available but there is some probability of getting some information. In case (i) for example, if a query "*Find an Indian restaurant 20 miles from here*" is asked from a ship in the middle of the ocean then the result will be null. We know that no restaurant can exist in the middle of the ocean; therefore, no information can be obtained. Thus, it is meaningless to ask this query. In case (ii) no information is available but the query "*List all traffic accidents 20 miles from here*" is likely to produce a meaningful result. We ask this query because we have some prior knowledge about the possibility of finding a list of accidents so such query is relevant for composing a *MinSet*. In this paper we do not investigated situation (a) we mainly focus our investigation on situation (b).

The situation (b) presents a common real-world scenario for which we develop scheme for composing the *MinSet*. If *MinSet* is already available then the discovery phase is not required and the identification can be initiated. Our idea of *MinSet* composition is based on the way we complete a jigsaw puzzle. In putting together all pieces correctly, we begin with a single piece and try to find another semantically matching piece (through reasoning) to attach to the first one. We then find and attach another piece that matches with the last two, and so on. We apply this "find and attach" step *intelligently* until the jigsaw is completed. A *knowledge base* will come into existence if we perform this step for every object. The following examples illustrate the application of "find and attach" approach using our built-in knowledge and experience to compose the *MinSet*.

Example 1: Information available*: video or photograph or a description of a chimney. We would like to answer the question "What object does this information represent?" No further information is available in the knowledge base so we need to build the MinSet and add it to the knowledge base. We apply "find and attach" approach as follows. Note that we know that a chimney is usually on the top of a house or of a small factory.*

a. *Shape and size of the chimney (useful SP to know if the chimney is a part of a house or a small factory. This is helpful but not conclusive.*
b. *Type and size of the roof (useful SP to know house roofs that are usually different than a factory's roof)*
c. *Location (useful SP to know if the area is residential or industrial)*
d. *Active (smoke coming out) or inactive (useful SP to correlate it with the time of year)*
e. *Weather (if active and it is summer then it is highly unlikely that the object is a house)*

We argue that with these parameters values we would be able to conclude that the object is a small factory or a house.

Example 2: Information available*: video or photograph or a description of a set of four wheels. We would like to answer the question "What object does this information represent?" We know that a set of wheels is an important property of vehicles so we begin looking into other vehicles properties.*

a. License plate (A useful SP to know if the object is a vehicle).
b. Contents of the place (A useful SP to confirm if it is a license plate for a vehicle.)
c. Location (A useful DP to know if this could be a possible location for a vehicle. A change in its value will indicate that it is a mobile object.)

In this case also we argue that with these *DP*s and *SP*s values we would be able to conclude that the object could be a vehicle. The discovery is dependent very much on our experience, logic and knowledge which are necessary to build a knowledge base for composition of *MinSet* of any object.

2.3 Graphical Representation of a Minimum Set (*MinSet*)

We represent our *MinSet* with a directed cyclic graph. *MinSet* = <*V, E*> where *V* is the set of vertices and *E* a set of weighted edges. A vertex represents a *DP* or *SP* and an edge connects a pair of *DP* or *SP*. The given information (Chimney) is called the *origin* vertex.

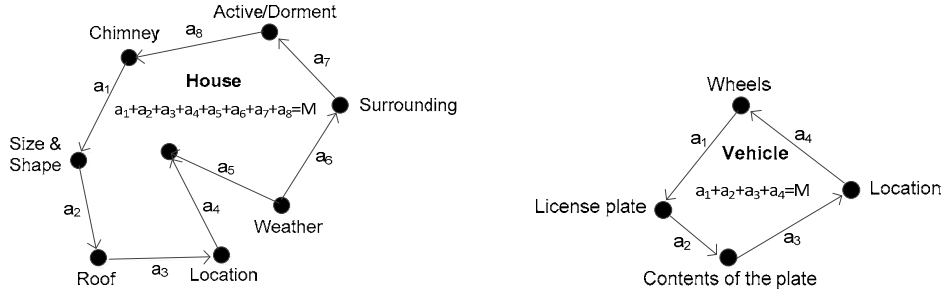


Figure 2. Graphical representation of *MinSet*s of a house and a vehicle

Figure 2 illustrates the *MinSet* Graph (MSG) of a *house* and a *vehicle* where Chimney and Wheels are the origin vertices. The weight of an edge represents the *affinity* (*aff*) of the connection between the two vertices. For example, the weight of the edge connecting *Chimney* and *Size and Shape* is a_1 and the weight of the edge connecting *Size and Shape* and *Roof* is a_2. If $a_1 > a_2$ then *Size and Shape* is a relatively more important (has higher value of *aff*) than *Roof* for discovering the object (House). There are two problems to resolve (a) recognize and include *DP*s and SPs of highest *aff* values and (b) define a termination condition or completion of a cycle.

Table 1. List of properties for creating *MinSet* for an object (Chimney)

Property Name	Root	Affinity (*aff*)	Domain
Shape	Y		
Size	Y		
Weight	Y		
Height	Y		
Color	Y		
Smell	N		
Location	Y		
Active	Y		
Dormant	Y		
Connected to	N		
Surrounding	N		
Weather	N		

Speed	N		
Temperature	Y		
Elevation	N		

3. Heuristic-based Algorithm DIIM (Discover, Identify, Introduce)

We discuss the development of our algorithm to create a *MinSet* for object discovery. We are motivated by Dijkstra's shortest path algorithm [1] to design our *MinSet* creation algorithm. The shortest path algorithm discovers the shortest path from a number of available paths. We apply this approach to create *MinSet* from a list of parameters. Thus, in shortest path algorithm the destination is known to the algorithm but in our algorithm a termination point needs to be defined.

The set of objects that an organization such as AFRL is interested is finite. Consequently the relevant sets of their *DP*s and *SP*s are also small. Our experience suggests that the set of properties required for discovering an object remains small. We identify a list of relevant *DP*s and *SP*s of objects (Table 1) that can be used intelligently to create a *MinSet*. These parameters are not sorted in any order. Table 2 lists partial information of objects that is received from the first responders or by some other means. We define a description called "Contextual index" that points to the *MinSet* of an object if there is one. If there is no *MinSet* present then it is created and added to this table. The Domain attribute of Table 1 identifies property domain. For example, the property *Height* of a *Chimney* has a domain that is defined by the city code. Similarly, the property *Temperature* has a domain that is possibly defined by the builder.

Table 2. List of Objects with their Contextual Indices.

Partial information	Contextual index
Chimney	Pointer to a set of predicates (edges) that refers to the object the Chimney represents
Wheels	Pointer to a set of predicates (edges) that refers to the object the Wheel represents
Doors	Pointer to a set of predicates (edges) that refers to the object the Door represents
License plate	Pointer to a set of predicates (edges) that refers to the object the License plate represents
Flag	Pointer to a set of predicates (edges) that refers to the object the Flag represents
------	

The *aff* value of a property is computed from the frequency of its use in objects discovery and the result. It can be computed with the following formula:

$$aff_n = aff_{n-1} + \sum_{i=1}^{n} \{^{-1=fail}_{+1=success} Oi$$

aff_n indicates the new value of affinity and aff_{n-1} denotes the value of last affinity of the property. The second term of this equation gives a value of 1 when the object (*Oi*) was successfully discovered -1 otherwise. Note that the *aff* value of a property will be low if it has been unsuccessfully used for a large number of objects.

We also define an *Ignore Factor (IF)*. Its value indicates when a property can be safely removed from the knowledge base. If the *aff* of a property remains zero for an extended period of time then the property can be safely removed (garbage collect) from the knowledge base.

3.1 Algorithm to Create a *MinSet* Tree from the Received Information

The algorithm creates the *MinSet* by incrementally adding the highest aff properties to the vertex. All properties with same aff values are added on the same level of the tree. We use example 1 given earlier to explain how our algorithm works.

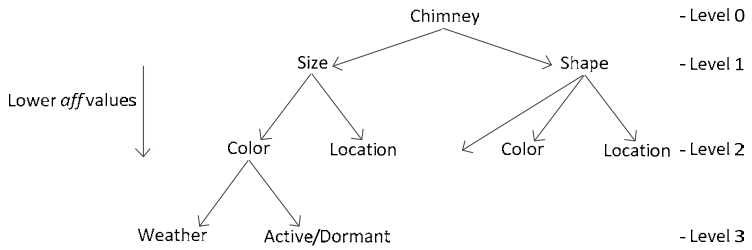


Figure 3. Building a *MinSet* for Chimney.

Example 1: Information available*: video or photograph or a description of a chimney. We would like to answer the question "What object does this information represent?" No further information is available in the knowledge base so we need to build the MinSet and add it to the knowledge base. We apply "find and attach" approach as follows. Note that we know that a chimney is usually on the top of a house or of a small factory.*

We assume that a *MinSet* does not exist in our knowledge base so it must be created from scratch. The information "Chimney" forms the root of the tree. We select a semantically related property from the table. From our experience we know that a chimney has a size that may determine the object it represents so we attach the size property to the root. Note that a random find and attach approach will eventually discover the object. Our aim is minimize *MinSet* creating time. Figure 3 sketches the steps.

A complete history of all earlier discoveries (*MinSet* trees) is maintained by DIIM system. This history is consulted after every "find and attach" step and in case of any similarity the tree building process is terminated.

3.2 Termination Condition for DIIM Algorithm

A termination condition is required to terminate the algorithm as soon as a *MinSet* is complete. Note that this condition is not necessary to complete the *MinSet*, it just optimizes the completion time. In the absence of a termination condition the algorithm is likely to attach all the properties of the table in the *MinSet*. The knowledge base helps to identify the completion of the *MinSet* of an object. The termination condition can be described through the following steps:

a. Pick and add one or more highest *aff* value properties to the root.
b. Search the knowledge base to find a set of predicates leading to the result. If not found and more properties are needed to attach then go to step a.
c. If the *MinSet* is complete then store this (graph) as a contextual index in the knowledge base (Table 2).

The entire knowledge base of an organization will contain predicates to cover all relevant objects the organization is interested in.

4. IDENTIFICATION AND INTRODUCTION

We briefly describe the identification and identification processes in this paper and report our complete investigation on these topics in our future work.

Identification process identifies the right object from a set of discovered objects. We introduce the scenario for introduction through a number of examples. We discuss the case when a brief high-level description about the object is received. This situation is relatively more complex and involves object identification and introduction. The discovery is given through a brief description of the object. The following example illustrates the situation.

Description 1: *In Khandahar, Afganistan, a terrorist cell may be entering from north at 10:00AM.*

Object discovery is not required because the object, terrorist cell, is available from the description. We need to identify uniquely the terrorist cell from many non-terrorist and other terrorist cells. The description indicates that the

object is entering from the north at 10:00AM. This helps us to concentrate on all those terrorist cells that are entering from north at 10:00AM. Now consider the following descriptions.

Description 2: *Track all terrorist cells in Khandahar, Afganistan*.
Description 3: *Track the terrorist cells that may cross the border of Khandahar*.
Description 4: *Track dangerous elements in Khandahar*.

Descriptions 2 and 3 do not need object discovery but 4 does because it is a command that implies a transfer of total discretion to the tracking system. That means first define a dangerous entity and then discover, identify, introduce and begin tracking.

We develop a tracker that can be appended to N-CET for object tracking. Information from local resources may come in any shape and size.

The information collected in identify step is fed to N-CET. There is a research issue here is: "What set of sensors must be assigned to capture the desired properties values for efficient tracking?" We assume that N-CET has multiple ports for connecting multiple sensors (video camera, night vision camera, radar, etc.) A human user may not be able to identify the most suitable set of sensors for object tracking. A sensor may create the *MinSet* but the system may select a different sensor for tracking the object. For example, a video camera inputs the minimum set but N-CET selects a radar sensor and RF scanner to track the object because of unfavorable weather conditions. The facility to dynamically select the most appropriate sensor during an active tracking is also highly desirable. Consider the situation where a video camera was tracking an object. If during tracking the weather changes and there is not enough light for the camera to take good picture then the system should automatically be able to switch to night vision camera to continue tracking. This of course will increase the data fusion complexity but it can be handled easily.

5. KNOWLEDGE BASE

Our knowledge base contains sets of predicates for object discovery and identification. A minimum subset of predicates must be satisfied to discover an object correctly. These predicates can be represented in textual form and also in graphical form. A relevant subset of predicates applied to the object properties that has two outcomes (a) minimum number of predicates are satisfied $\Rightarrow$ object identified and (b) less (or none) than minimum number of predicates satisfied $\Rightarrow$ input sample is too small (information about the object) to identify the object. The following example may help to understand the point.

Example 1: User input data – License plate number: OQX 009.
It is hard to answer the question: Is it a car?
Is it a van?
Is it a motorbike?

Example 2: Input data – L/L: 38/35
Physical dimension and shape are given
It is hard to answer the question: Is it an apartment or a bank or a house?

Example 3: User input data – License plate #: OQX 009, # of wheels = 4, length = XY inches, sunroof = Y, # of doors = 4, windshield wiper on back window = N.
Identification result: a car (very likely).

Examples 1 and 2 do not have *MinSet* input but Example 3 has. A *MinSet* of an object can either be composed by a user or could be initiated by the user and completed by the system.

A predicate may look like when the photo of a Chimney is available

If the Size (Chimney) is [x, y] then
If Shape (Chimney) is [a, b] then
If Height (Chimney) is [m, n] then
Location (Chimney) is [Li, Lj] then Result = the Chimney represents a house

Else ---
Else ---
Else---

A set of sensors will then be deployed to measure these parameters to discover the object the Chimney represents. In our knowledge base these set of predicates are represented in closed cyclic graph. A contextual index is generated for such graphs. Some of the ways N-CET could make use of knowledge base are:

a. User composes the *MinSet* of an object from information received through local resources such as people, police, etc., which is input to N-CET. The user can be the pilot of the carrier of N-CET node, ground control, etc. This means that the *MinSet* can be input to the system remotely also.
b. System itself composes a set of profile for objects for tracking. This facility is useful for tracking/monitoring the movement of terrorist groups in a hard to reach terrain.

6. CONCLUSIONS

In this paper we investigated object tracking from information management view point. We formally defined a number of important concepts such as sensor, stream data, data capture, etc. We introduced and formally defined three connected processes called discovery, identification, and introduction and used them to represent the commonly know steps Find $\rightarrow$ Fix $\rightarrow$ Track $\rightarrow$ Target engage. The focus of this paper was object discovery process leaving the other two processes (identification and introduction) for our future work. We developed the notion of Minimum Set (*MinSet*) that is composed of dynamic and static property set of the object for object discovery. We introduced a construct called "contextual index" that points to a *MinSet* graph in the knowledge base. The introduction of *MinSet* graph was mainly to optimize the discovery process. If a contextual index is not present in the knowledge base then the system builds one for the object from the partial information available from a source (first responder, a video or something else.) Our future works will report our schemes for object identification, introduction, and accurate selection of a set of sensors for object tracking.

References

[1] Dijkstra, E., "A note on two problems in connexion with graphs," *Numerische Mathematik* **1**, 269–271 (1959).
[2] Metzler, James, Linderman, Mark, and Seversky, Lee., "N-CET: Network centric exploitation and tracking," MILCOM Conference, Boston World Trade Center, Boston, MA 2009.
[3] Linderman, M., Brichacek, J., Haines, S., Siegel, B., Chase, G., Ouellet, D. and O'May, J., "A reference model for information management to support coalition information sharing needs," *The 10th International Command and Control Technology Symposium (ICCRTS05)*, (2005).